

Tutorial on Measuring Performance Counters

Hadi Asghari Moghaddam

In every CPU there are several performance counters in each core that you can measure different events with them.

Depends on the CPU type; there is different number of counters and also different events that you can measure them. In Intel CPUs usually there are 3 fixed counters per core and also 4 general purpose counters per logical core. For the fixed counters you can just measure the pre-fixed events, but for general purpose counters you have a list of different events that contain “Architectural Events” and “Non-Architectural Events”, in which one of them is common in all Intel architectures and the other one varies according to the generation and model.

Non-Architectural counters: Supports events for monitoring performance using counting or sampling usage. These events vary from one processor model to another. These events for a given microarchitecture cannot be enumerated using CPUID.

Architectural Performance Monitoring: This class supports the same counting and sampling usage with a smaller set of available events. The visible behavior of an architectural performance event is consistent across processor implementation. Availability of architectural performance monitoring capabilities is enumerated using CPUID.0AH.

For reading the MSR values “rdmsr” command should be used. “rdmsr” is using the ECX parameter as its argument by storing its value in register ECX, the result of the “rdmsr” operation is stored in register EAX:EDX.

Following table is the address and description of the fixed MSRs that can be used.

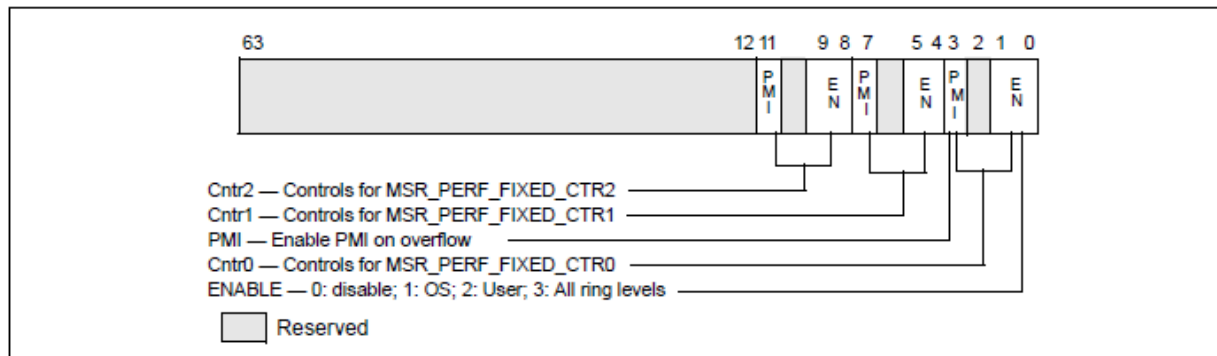
Event Name	Fixed-Function PMC	PMC Address
INST_RETIRED.ANY	MSR_PERF_FIXED_CTR0/IA32_FIXED_CTR0	309H
CPU_CLK_UNHALTED.CORE	MSR_PERF_FIXED_CTR1/IA32_FIXED_CTR1	30AH
CPU_CLK_UNHALTED.REF	MSR_PERF_FIXED_CTR2/IA32_FIXED_CTR2	30BH

For accessing these counters, there are several registers that you should control and access the above addresses via them:

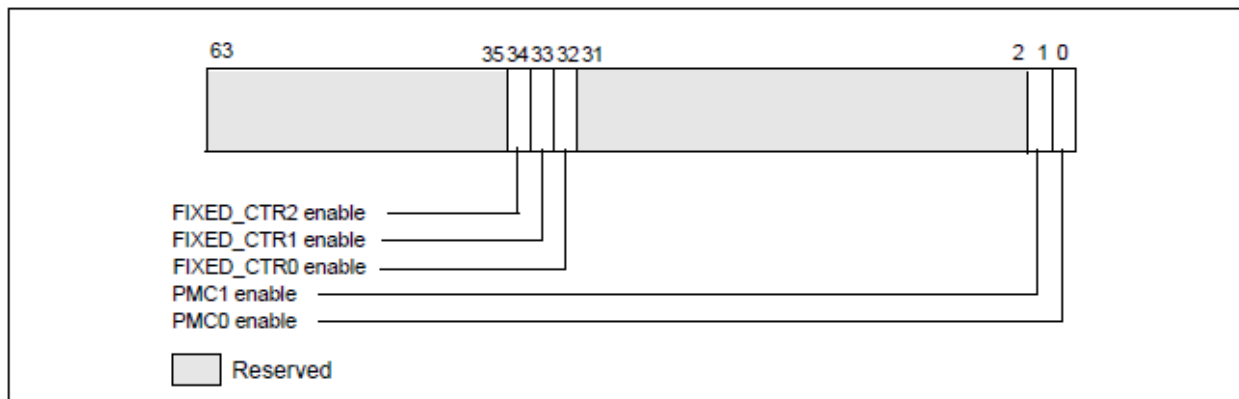
- MSR_PERF_FIXED_CTR_CTRL MSR
- MSR_PERF_GLOBAL_CTRL MSR
- MSR_PERF_GLOBAL_STATUS MSR
- MSR_PERF_GLOBAL_OVF_CTRL MSR

Detail of their usage and the meaning of bits can be accessed in [1], but for a brief introduction following is the demonstration of each of them.

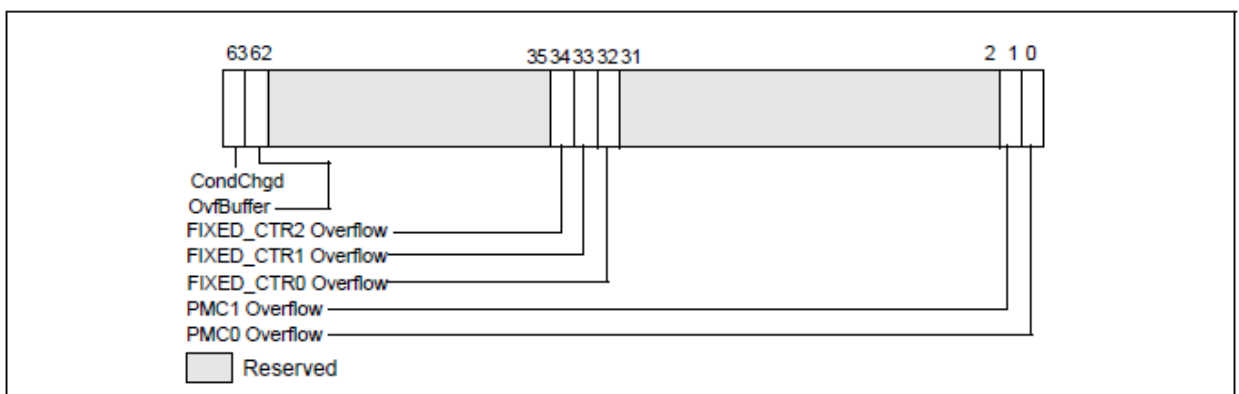
MSR_PERF_FIXED_CTRL MSR:



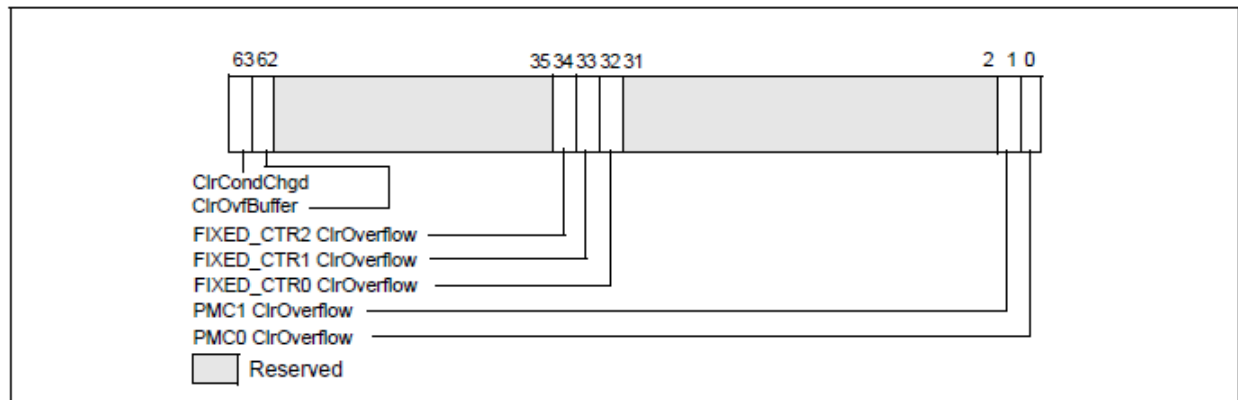
MSR_PERF_GLOBAL_CTRL MSR:



MSR_PERF_GLOBAL_STATUS MSR:

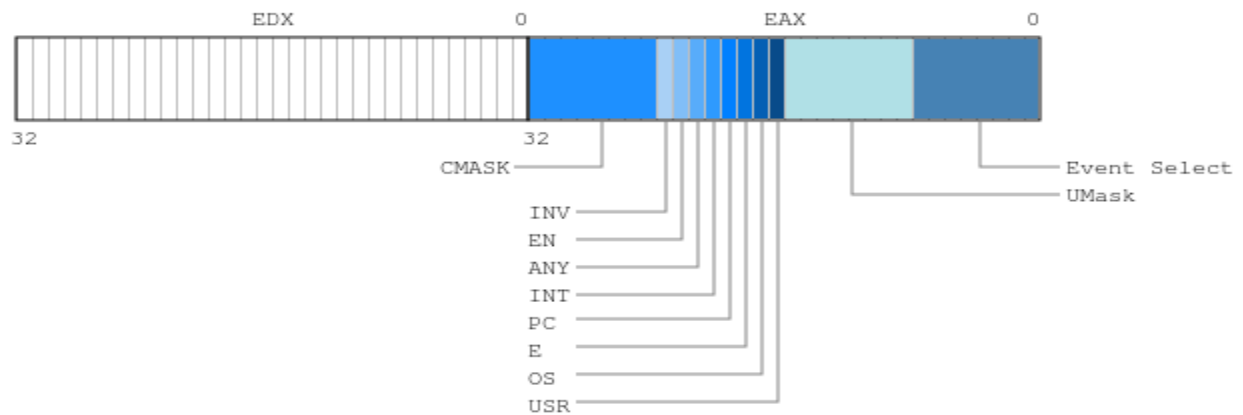


MSR_PERF_GLOBAL_OVF_CTRL MSR:



General Purpose Counters:

For general purpose counters you have 4 IA32_PERFEVTSELx registers for configuring each counter based on your desired event and access type:



Each event has an event select and an umask value, you should find your desired events from [1] based on your machine's cpu generation!

After configuring each IA32_PERFEVTSELx you can read the counter values on IA32_PMCx registers.

Register Address		Architectural MSR Name and bit fields (Former MSR Name)	MSR/Bit Description	Introduced as Architectural MSR
Hex	Decimal			
C1H	193	IA32_PMC0 (PERFCTR0)	General Performance Counter 0 (R/W)	If CPUID.0AH: EAX[15:8] > 0
C2H	194	IA32_PMC1 (PERFCTR1)	General Performance Counter 1 (R/W)	If CPUID.0AH: EAX[15:8] > 1
C3H	195	IA32_PMC2	General Performance Counter 2 (R/W)	If CPUID.0AH: EAX[15:8] > 2
C4H	196	IA32_PMC3	General Performance Counter 3 (R/W)	If CPUID.0AH: EAX[15:8] > 3
C5H	197	IA32_PMC4	General Performance Counter 4 (R/W)	If CPUID.0AH: EAX[15:8] > 4
C6H	198	IA32_PMC5	General Performance Counter 5 (R/W)	If CPUID.0AH: EAX[15:8] > 5
C7H	199	IA32_PMC6	General Performance Counter 6 (R/W)	If CPUID.0AH: EAX[15:8] > 6
C8H	200	IA32_PMC7	General Performance Counter 7 (R/W)	If CPUID.0AH: EAX[15:8] > 7

186H	390	IA32_PERFVTSELO (PERFVTSELO)	Performance Event Select Register 0 (R/W)	If CPUID.0AH: EAX[15:8] > 0
		7:0	Event Select: Selects a performance event logic unit.	
		15:8	UMask: Qualifies the microarchitectural condition to detect on the selected event logic.	
		16	USR: Counts while in privilege level is not ring 0.	
		17	OS: Counts while in privilege level is ring 0.	
		18	Edge: Enables edge detection if set.	
		19	PC: enables pin control.	
		20	INT: enables interrupt on counter overflow.	
		21	AnyThread: When set to 1, it enables counting the associated event conditions occurring across all logical processors sharing a processor core. When set to 0, the counter only increments the associated event conditions occurring in the logical processor which programmed the MSR.	
		22	EN: enables the corresponding performance counter to commence counting when this bit is set.	
		23	INV: invert the CMASK.	

		31:24	CMASK: When CMASK is not zero, the corresponding performance counter increments each cycle if the event count is greater than or equal to the CMASK.	
		63:32	Reserved.	
187H	391	IA32_PERFEVTSEL1 (PERFEVTSEL1)	Performance Event Select Register 1 (R/W)	If CPUID.0AH: EAX[15:8] > 1
188H	392	IA32_PERFEVTSEL2	Performance Event Select Register 2 (R/W)	If CPUID.0AH: EAX[15:8] > 2
189H	393	IA32_PERFEVTSEL3	Performance Event Select Register 3 (R/W)	If CPUID.0AH: EAX[15:8] > 3

[1] Intel® 64 and IA-32 Architectures Software Developer's Manual